# Quadrature and Tridiagonal Matrices

Gunnar Golden, Hannah Leopold-Brandt, Kyle Monette

December 7, 2023

# Introduction

Here we consider *symmetric and unreduced tridiagonal matrices*

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{bmatrix}.$$

We'll introduce *Gaussian Quadrature* and explain how it is related the eigenvalue problem of $T$.

# Quadrature – Basic Ideas

The idea of (interpolary) quadrature is to approximate a definite integral of a function using a polynomial interpolant:

$$\int_a^b f(x)\,dx \quad \approx \quad \int_a^b p_n(x)\,dx$$

where $p_n$ is a polynomial of degree at most $n$ that interpolates $f$ at $n+1$ points in $[a, b]$.

Question: Can we do this without constructing $p_n$?

Question: For what functions $f$ is this "exact"?

# Gaussian Quadrature Rules

There are many different types of quadrature rules, depending on the functions $f$ you wish to integrate:

Gauss-Legendre: $$\int_{-1}^{1} f(x) \cdot 1 \, dx$$

Gauss-Chebyshev: $$\int_{-1}^{1} f(x) \cdot \frac{1}{\sqrt{1-x^2}} \, dx$$

Gauss-Laguerre: $$\int_{0}^{\infty} f(x) \cdot e^{-x} \, dx$$

Gauss-Hermite: $$\int_{-\infty}^{\infty} f(x) \cdot e^{-x^2} \, dx$$

The difference being what *weight function* $w(x)$ you choose.

We'll concentrate on Gauss-Legendre here.

# Interpolating Functions

Given a function $f(x)$ and $n+1$ **distinct** nodes $\{x_0, \ldots, x_n\} \in [a, b]$, we can *interpolate* $f$ by a $n$ degree polynomial $p_n(x)$

$$p_n(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x)$$

where $\ell_j(x)$ is the $j$-th Lagrange basis function:

$$\ell_j(x) = \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k}, \quad 0 \leq j \leq n \qquad \ell_j(x_k) = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

Why?

$$\begin{aligned} p_n(x_k) &= \sum_{j=0}^{n} f(x_j)\ell_j(x_k) \\ &= f(x_0)\ell_0(x_k) + \cdots + f(x_k)\ell_k(x_k) + \cdots + f(x_n)\ell_n(x_k) \\ &= f(x_k) \end{aligned}$$

# Interpolating Functions

Last Slide:

$$p_n(x) \quad = \quad \sum_{j=0}^{n} f(x_j)\ell_j(x)$$

$$\int_a^b p_n(x)\,dx \quad = \quad \int_a^b \sum_{j=0}^{n} f(x_j)\ell_j(x)\,dx \quad = \quad \sum_{j=0}^{n} f(x_j)\int_a^b \ell_j(x)\,dx.$$

This defines the weights of the quadrature rule:

$$w_j = \int_a^b \ell_j(x)\,dx, \qquad 0 \le j \le n+1.$$

Therefore:

$$\int_a^b f(x)\,dx \quad \approx \quad \sum_{j=0}^{n} w_j\, f(x_j).$$

# Orthogonal Polynomials

## Definition 1 (Orthogonal Polynomials)

With a weight function $w(x)$ that is nonnegative and continuous on $(a, b)$, we define the *weighted inner product*

$$\langle f(x), g(x) \rangle = \int_a^b f(x)g(x)w(x)\,dx.$$

Two polynomials $p$ and $q$ are said to be *orthogonal* if $\langle p(x), q(x) \rangle = 0$.

We will use $(a, b) = (-1, 1)$ and $w(x) = 1$ (for Gauss-Legendre quadrature).

# Building Up GL Quadrature

Given orthogonal polynomials $\phi_0(x) = 1$, and $\phi_1(x) = x$, then apply Gram-Schmidt with the inner product above (*any* weight function) to obtain the recurrence

$$\phi_{k+1}(x) = (x - \alpha_{k+1})\phi_k(x) - \sqrt{\beta_k}\phi_{k-1}(x), \qquad k = 0, \ldots, n, \quad \phi_{-1}(x) = 0.$$

This generates a set of *pairwise orthogonal polynomials*

$$\{\phi_0, \phi_1, \ldots, \phi_n, \phi_{n+1}\}.$$

One can show that the constants $\alpha_k$ and $\beta_k$ satisfy

$$\alpha_k = \frac{\langle x\phi_{k-1}, \phi_{k-1}\rangle}{\langle \phi_{k-1}, \phi_{k-1}\rangle}, \quad 1 \le k \le n+1, \qquad \beta_k = \frac{\langle x\phi_k, \phi_{k-1}\rangle}{\langle \phi_{k-1}, \phi_{k-1}\rangle}, \quad 1 \le k \le n$$

where we separately define $\beta_0 = \langle 1, 1\rangle$.

# Building Up GL Quadrature

Last Slide:

$$\alpha_k = \frac{\langle x\phi_{k-1}, \phi_{k-1}\rangle}{\langle \phi_{k-1}, \phi_{k-1}\rangle}, \qquad \beta_k = \frac{\langle x\phi_k, \phi_{k-1}\rangle}{\langle \phi_{k-1}, \phi_{k-1}\rangle}$$

For GL quadrature, importantly we find

$$\alpha_k = 0 \quad \forall\, k, \qquad \beta_k = \frac{k^2}{4k^2 - 1}, \quad 1 \le k \le n, \qquad \beta_0 = 2.$$

In fact, the polynomials obtained are the *Legendre polynomials*:

$$\phi_0(x) = 1, \quad \phi_1(x) = x, \quad \phi_2(x) = x^2 - \frac{1}{3}, \quad \phi_3(x) = x^3 - \frac{3}{5}x, \quad \ldots.$$

...OK, so what?

# Determining the Nodes

One can show that the approximation

$$\int_a^b f(x)\,dx \quad \approx \quad \int_a^b p_n(x)\,dx$$

is exact for $f(x)$ being a polynomial of degree at most $2n+1$ **if** the nodes are chosen to be the roots of $\phi_{n+1}(x)$.

## Lemma 2

*The polynomial $\phi_{n+1}(x)$ has $n+1$ distinct real roots in $[a, b]$.*

In Gauss-Legendre Quadrature, recall that the $\phi$ functions are the Legendre polynomials — functions which finding roots of is an inefficient and numerically unstable task.

## Theorem 3

*Given the set of orthogonal polynomials* $\{\phi_0(x), \ldots, \phi_{n+1}(x)\}$*, then* $\lambda$ *is a root of* $\phi_{n+1}(x)$ *if and only if* $\lambda$ *is an eigenvalue of the matrix*

$$J_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \sqrt{\beta_n} \\ & & & \sqrt{\beta_n} & \alpha_n \end{bmatrix},$$

*where the associated eigenvector is*

$$v(\lambda) = \begin{bmatrix} \phi_0(\lambda) \\ \phi_1(\lambda)/\sqrt{\beta_1} \\ \phi_2(\lambda)/\sqrt{\beta_1\beta_2} \\ \vdots \\ \phi_n(\lambda)/\sqrt{\beta_1 \ldots \beta_n} \end{bmatrix}.$$

# Eigenvectors

Golub and Welsch proved that the weights $w_j$ can be found by

$$w_j = \frac{\beta_0}{\|v_j\|_2^2}$$

where $v_j$ is the $j$-th eigenvector of $J_n$.

# In Summary

In conclusion, for GL Quadrature:

1. Create the set of Legendre polynomials $\{\phi_0, \ldots, \phi_{n+1}\}$ and calculate $\alpha_k$ and $\beta_k$.
2. Create the tridiagonal matrix $J_n$.
3. Find the eigenvalues of $J_n$, which are the nodes $x_j$ used in the interpolation.
4. Find the eigenvectors of $J_n$, which give us the quadrature weights $w_j$.

$$\int_{-1}^{1} f(x)\, dx \quad \approx \quad \sum_{j=0}^{n} w_j f(x_j)$$

Why choose the nodes and weights in this way?
Because this approximation is exact (!!) for any degree $2n+1$ polynomial $f(x)$.

# Example

For $n = 4$, the Legendre polynomials are

$$\phi_0(x) = 1 \qquad \phi_1(x) = x \qquad \phi_2(x) = x^2 - \frac{1}{3}$$
$$\phi_3(x) = x^3 - \frac{3}{5}x \qquad \phi_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}$$

The matrix $J$ is

$$J_4 = \begin{bmatrix} 0 & 0.5774 & & & \\ 0.5774 & 0 & 0.5164 & & \\ & 0.5164 & 0 & 0.5071 & \\ & & 0.5071 & 0 & 0.5040 \\ & & & 0.5040 & 0 \end{bmatrix}.$$

The eigenvalues of $J_4$ are the nodes $x_0, x_1, x_2, x_3, x_4$ and the weights are found by the Golub and Welsh formula.

# Example

With $n = 4$, we can exactly integrate polynomials of degree $\leq 2n + 1 = 9$:

$$\int_{-1}^{1} x^9 + x^6 \, dx \quad = \quad \sum_{k=0}^{4} w_j \underbrace{f(\lambda_j)}_{f(x_j)} \quad = \quad 0.2857$$

$$\int_{-1}^{1} x^{12} \, dx \qquad \text{Error:} \quad 0.008$$

$$\underbrace{\int_{-1}^{1} \sin(e^{x^2}) \, dx}_{\text{NOT integrable!}} \qquad \text{Error:} \quad 9.9 \times 10^{-4}$$

# Eigenvalues of Tridiagonal Matrix

But *how* did we find the eigenvalues of the matrix in the example?

$$J_4 = \begin{bmatrix} 0 & 0.5774 & & & \\ 0.5774 & 0 & 0.5164 & & \\ & 0.5164 & 0 & 0.5071 & \\ & & 0.5071 & 0 & 0.5040 \\ & & & 0.5040 & 0 \end{bmatrix}$$

We need another solution that does not use the Theorem. That is, NOT by finding the roots of $\phi_{n+1}$.

# QR Algorithm

The Solution? *QR* Algorithm

**Algorithm 1** *QR*-Algorithm

```
A^(0) = A
for k = 1, 2, ... do
    Q^(k) R^(k) = A^(k-1)
    A^(k) = R^(k) Q^(k)
end for
```

- Recall that the *QR* algorithm converges to the Schur form of $A$.
  - $A = PTP^T$ where $T$ upper triangular and $P$ orthogonal.
- When $A$ is *symmetric*, then $A^{(k)}$ converges to a diagonal matrix!
- Due to similarity, we know the eigenvalues of $A$.

# QR Algorithm

Note that Gauss-Legendre matrices $J_n$ have a zero diagonal.

This is a problem!

Recall the convergence theorem from NLA:

**Theorem 4 (NLA Theorem 28.4)**

*If the QR algorithm is applied to a real symmetric matrix with eigenvalues satisfying $|\lambda_1| > \cdots > |\lambda_n|$ and Q has nonsingular leading principal minors, then $A^{(k)}$ converges.*

The eigenvalues of $J_4$ are

$$-0.9062, \ -0.5385, \ 0, \ 0.5385, \ 0.9062.$$

Which are not strictly monotonic in absolute value.

Therefore, the $QR$ algorithm on $J_n$ does not converge.

# The Solution

We must shift these matrices first before applying $QR$ algorithm:

$$J_4 + I = \begin{bmatrix} 1 & 0.5774 & & & \\ 0.5774 & 1 & 0.5164 & & \\ & 0.5164 & 1 & 0.5071 & \\ & & 0.5071 & 1 & 0.5040 \\ & & & 0.5040 & 1 \end{bmatrix}.$$

Now the QR algorithm above will converge, and we can recover the eigenvalues.

**Lemma 5**

*If $A + I$ has an eigenvalue $\lambda$, then $A$ has an eigenvalue $\lambda - 1$.*

# QR Algorithm

Eigenvalues of the shifted matrix

$$J_4 + I = \begin{bmatrix} 1 & 0.5774 & & & \\ 0.5774 & 1 & 0.5164 & & \\ & 0.5164 & 1 & 0.5071 & \\ & & 0.5071 & 1 & 0.5040 \\ & & & 0.5040 & 1 \end{bmatrix}$$

are given by

$$0.0938,\ 0.4615,\ 1,\ 1.5385,\ 1.9062.$$

Therefore $J_n$ has eigenvalues

$$-0.9062,\ -0.5385,\ 0,\ 0.5385,\ 0.9062.$$

# Conclusion

Last Slide: $J_n$ has eigenvalues

$$-0.9062, \ -0.5385, \ 0, \ 0.5385, \ 0.9062.$$

Then the eigenvectors are formed by Golub-Welsch. For example,

$$v_1 = \begin{bmatrix} \phi_0(-0.9062) \\ \phi_1(-0.9062)/\sqrt{\beta_1} \\ \phi_2(-0.9062)/\sqrt{\beta_1\beta_2} \\ \phi_3(-0.9062)/\sqrt{\beta_1\beta_2\beta_3} \\ \phi_4(-0.9062)/\sqrt{\beta_1\beta_2\beta_3\beta_4} \end{bmatrix} = \begin{bmatrix} 1 \\ -1.5695 \\ 1.6362 \\ -1.3256 \\ 0.7372 \end{bmatrix}$$

which implies

$$w_1 = \frac{\beta_0}{\|v_1\|_2^2} = \frac{2}{8.4414} = 0.2369$$

Questions?

# References


- Embree, Mark. *Lecture Notes on Numerical Analysis*, 2016. personal.math.vt.edu/embree/math5466/nanotes.pdf
- Trefethen, Lloyd N., and David Bau. Numerical linear algebra. Vol. 181. Siam, 2022.
- Parlett, Beresford N. *The Symmetric Eigenvalue Problem*, 1998